

FIG. 1

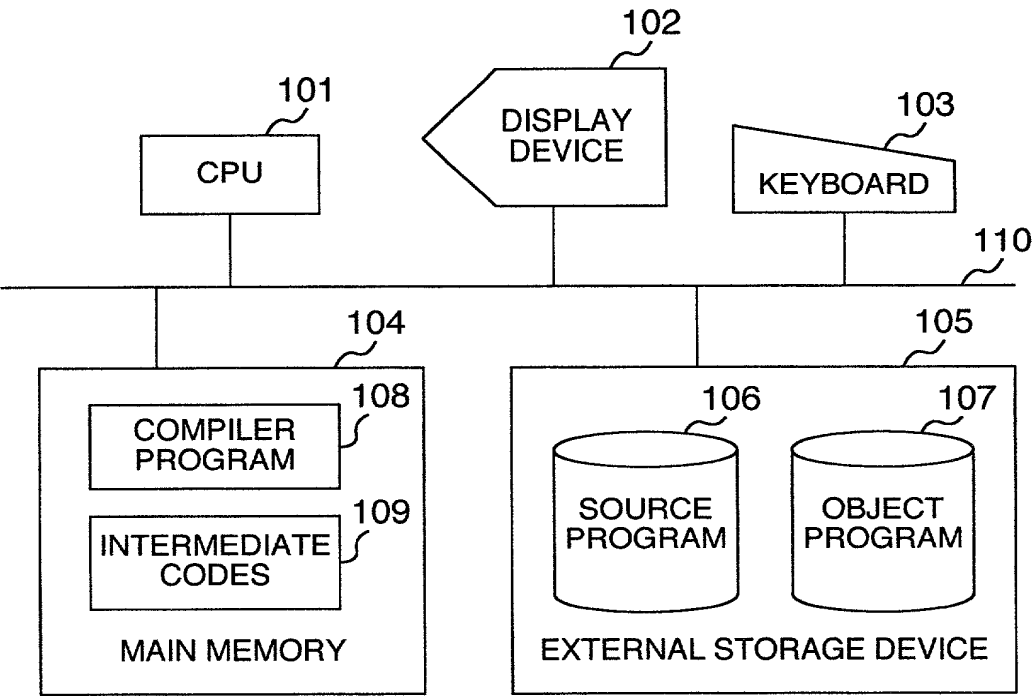


FIG. 2

```
200
while (cond) {           (201)
    c = a+b;             (202)
    *p = c;              (203)
    p = p->next;         (204)
}                         (205)
```

# FIG. 3

300

while (cond) {	(301)
ld  r1=[&a]	(302)
ld  r2=[&b]	(303)
add r3=r1,r2	(304)
st  *p=r3	(305)
p = p->next	(306)
}	(307)

# FIG. 4

400

ld.a  r1=[&a]	(401)
ld.a  r2=[&b]	(402)
add  r3=r1,r2	(403)
while (cond) {	(404)
chk.a  r1,recover1	(405)
L1:  chk.a  r2,recover2	(406)
L2:  st  *p=r3	(407)
p = p->next	(408)
}	(409)
recover1:	(410)
ld.a  r1=[&a]	(411)
add  r3=r1,r2	(412)
br  L1	(413)
recover2:	(414)
ld.a  r2=[&b]	(415)
add  r3=r1,r2	(416)
br  L2	(417)

FIG. 5

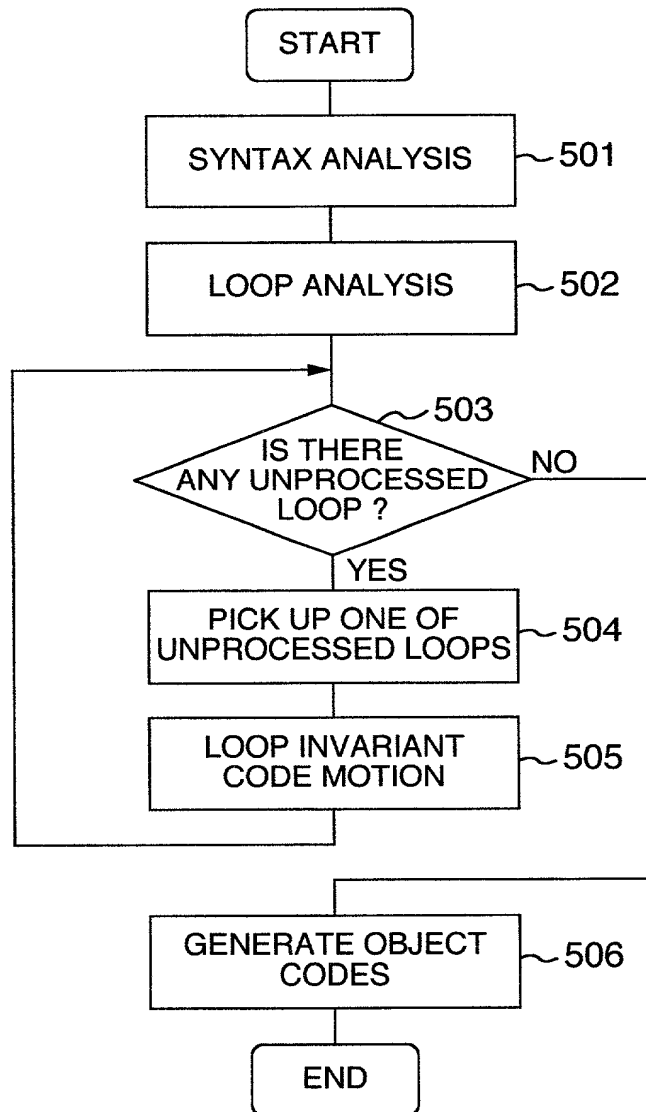


FIG. 6

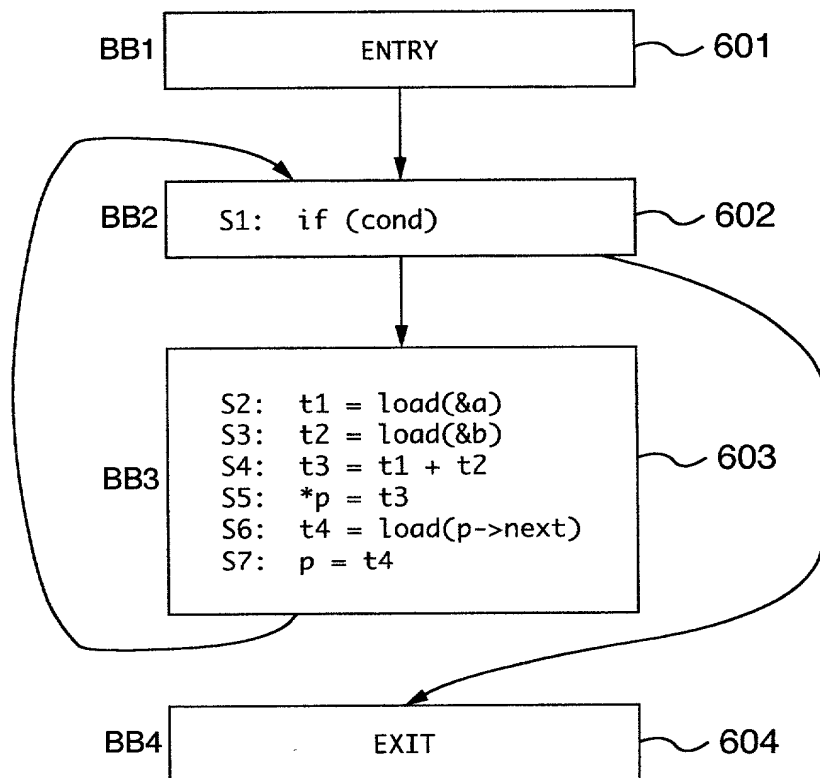


FIG. 7

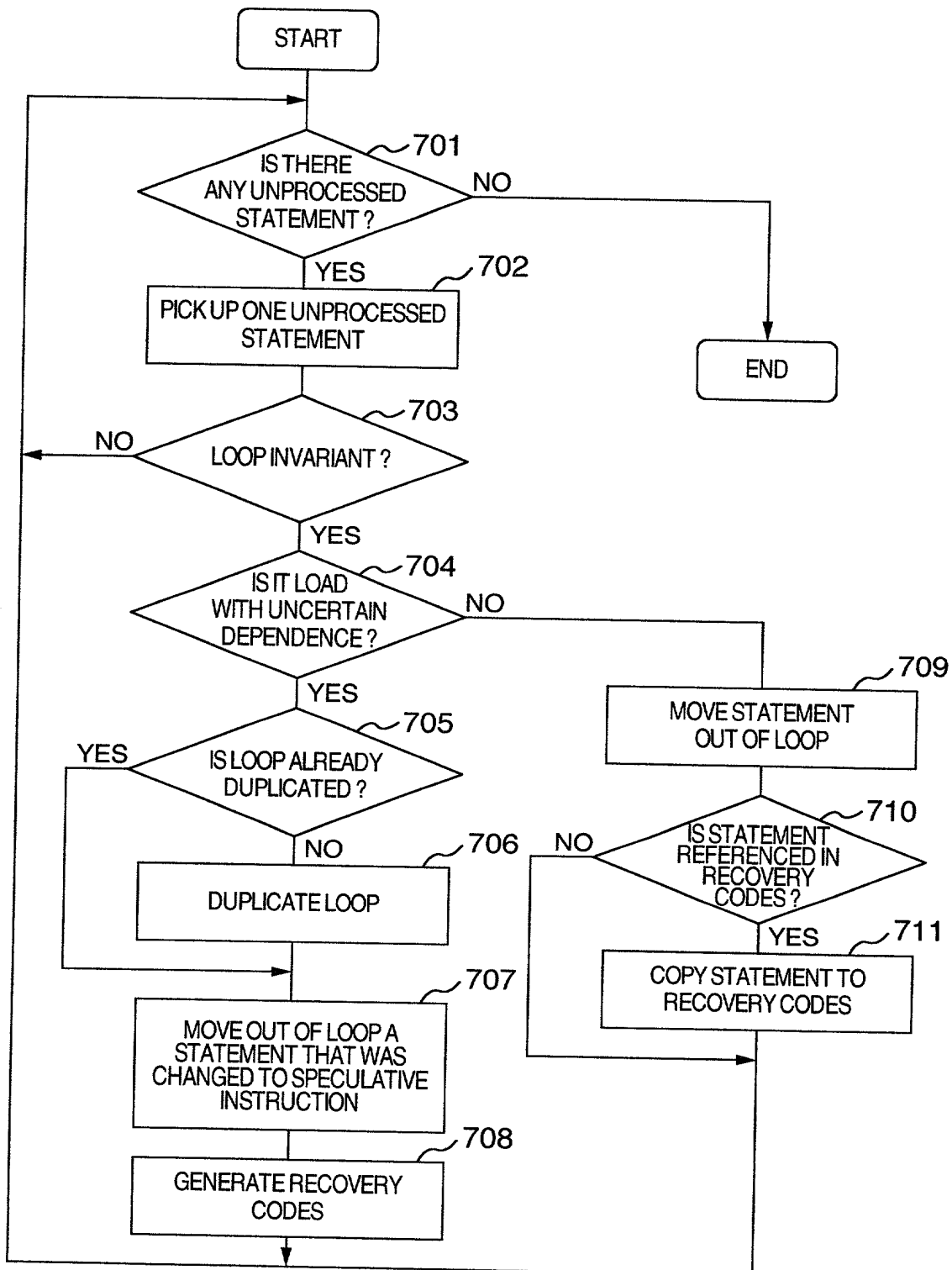


FIG. 8

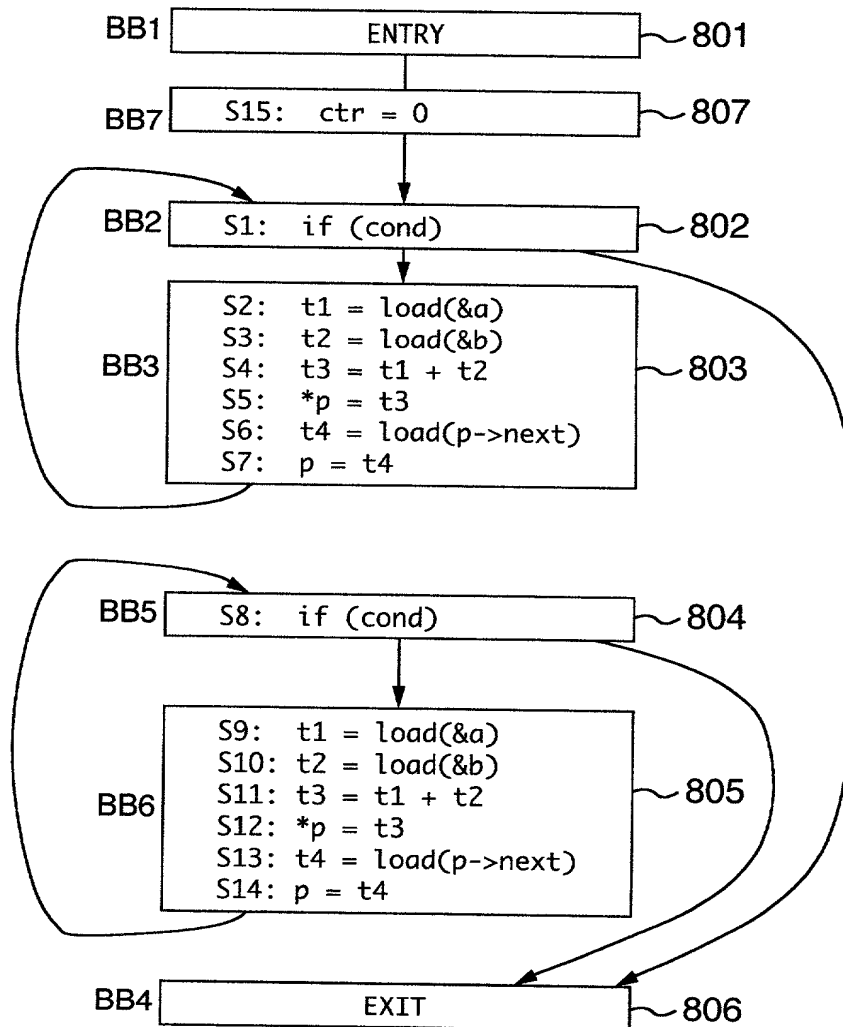
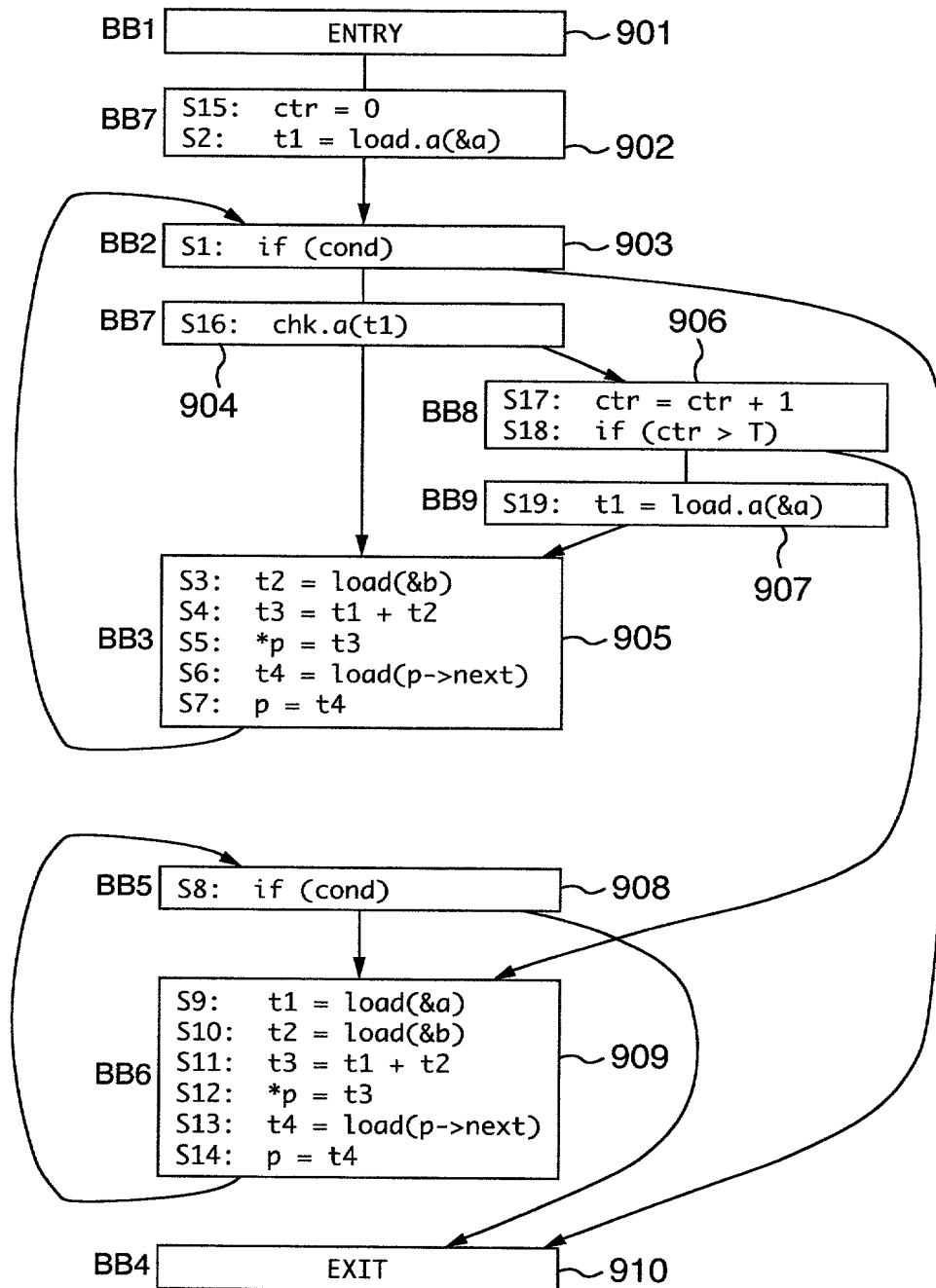
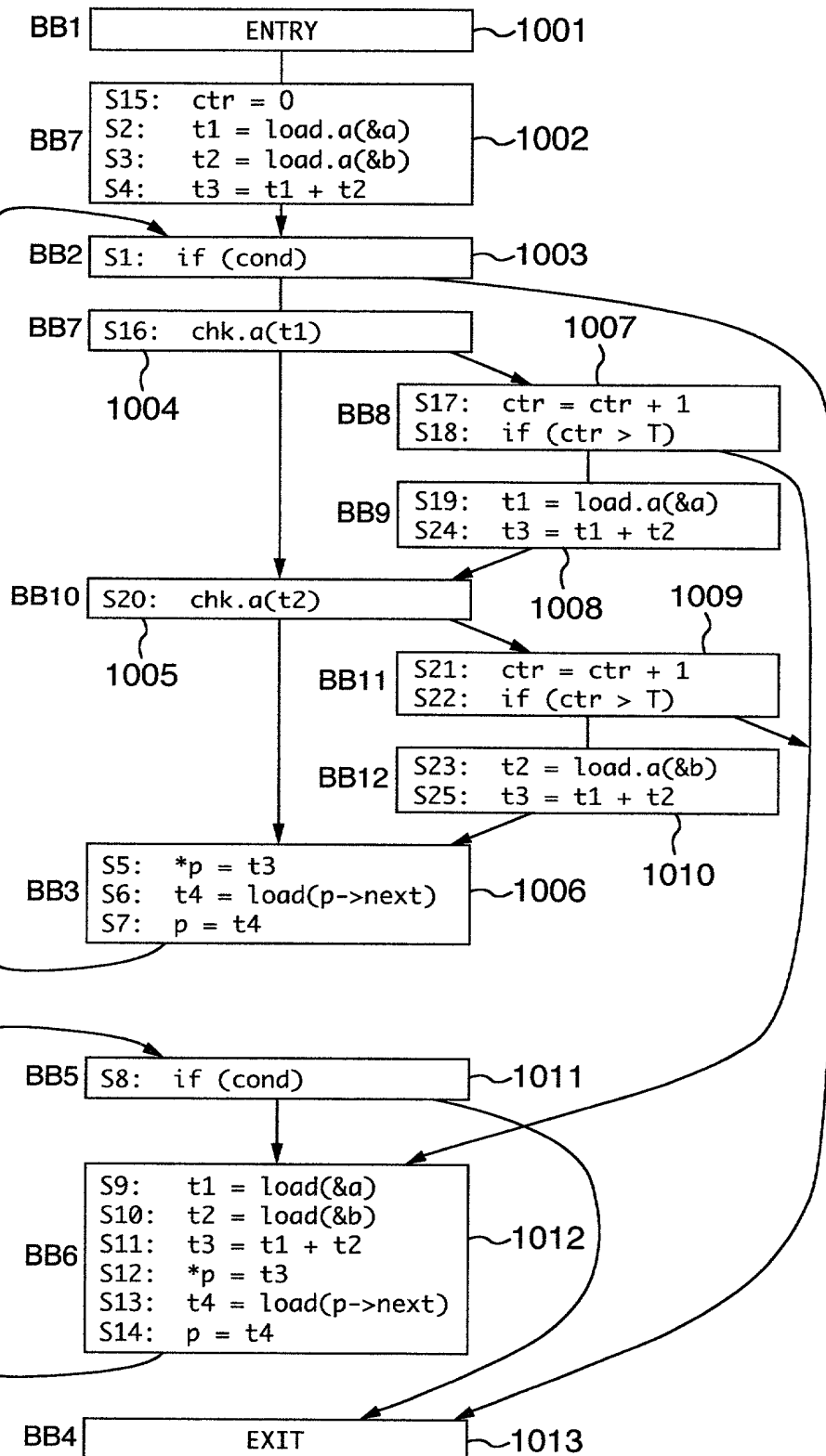


FIG. 9



# FIG. 10





# FIG. 11

1100

```

copy r4=0 (1101)
ld.a r1=[&a] (1102)
ld.a r2=[&b] (1103)
add r3=r1,r2 (1104)
while (cond) { (1105)
    chk.a r1,recover1 (1106)
L1: chk.a r2,recover2 (1107)
L2: st *p=r3 (1108)
    p = p->next (1109)
} (1110)
goto exit (1111)

while (cond) { (1112)
L3: ld r1=[&a] (1113)
    ld r2=[&b] (1114)
    add r3=r1,r2 (1115)
    st *p=r3 (1116)
    p = p->next (1117)
} (1118)
exit: (1119)

-----

recover1: (1120)
add r4=r4,1 (1121)
bc r4>T,L3 (1122)
ld.a r1=[&a] (1123)
add r3=r1,r2 (1124)
br L1 (1125)

-----

recover2: (1126)
add r4=r4,1 (1127)
bc r4>T,L3 (1128)
ld.a r2=[&b] (1129)
add r3=r1,r2 (1130)
br L2 (1131)

```



# FIG. 13

1300

```

copy   r4=0                (1301)
copy   r5=0                (1302)
ld.a   r1=[&a]              (1303)
ld.a   r2=[&b]              (1304)
add     r3=r1,r2            (1305)
while (cond) {              (1306)
    add     r5=r5,1          (1307)
    chk.a   r1,recover1      (1308)
L1: chk.a   r2,recover2      (1309)
L2: st      *p=r3            (1310)
        p = p->next          (1311)
}                                (1312)
goto exit                    (1314)

while (cond) {              (1315)
L3: ld      r1=[&a]          (1316)
        ld      r2=[&b]          (1317)
        add     r3=r1,r2        (1318)
        st      *p=r3          (1319)
        p = p->next          (1320)
}                                (1321)
exit:                                (1322)
-----
recover1:                    (1323)
add     r4=r4,1              (1324)
div      r6=r4,r5            (1325)
bc       r6>T,L3             (1326)
ld.a     r1=[&a]              (1327)
add      r3=r1,r2            (1328)
br       L1                  (1329)
-----
recover2:                    (1330)
add      r4=r4,1              (1331)
div      r6=r4,r5            (1332)
bc       r6>T,L3             (1333)
ld.a     r2=[&b]              (1334)
add      r3=r1,r2            (1335)
br       L2                  (1336)

```